AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for managing heap memory in a multitasking system, comprising:

reserving a guaranteed amount of heap memory for a task from a common heap in the multitasking system, wherein the heap memory reserved for the task is separate from heap memory reserved for all other tasks in the common heap of the multitasking system;

receiving a request from the task to allocate heap memory for a new object; and

if heap memory is available in the guaranteed amount of heap memory for the task, allocating heap memory for the new object from the guaranteed amount of heap memory;

if heap memory is not available, checking if surplus heap memory is available in the common heap;

if surplus heap memory is available ~~so~~, reserving an additional amount of heap memory to the task from the common heap and allocating heap memory for the new object from the additional amount of heap memory, whereby allocating heap memory for the new object from the additional amount of heap memory delays garbage collection; and

if surplus heap memory is not available, performing garbage collection on the heap memory reserved for the task, wherein the task space is separate from all other task ~~space~~ spaces; ~~whereby~~

wherein other tasks continue normal execution without interruption due to garbage collection~~, or subsequent memory compaction~~.

2. (Previously presented) The method of claim 1, wherein if surplus heap memory is not available in the common heap in addition to heap memory allocated to tasks, the method further comprises

performing garbage collection on heap memory to reclaim unused reserved heap memory, and

allocating heap memory for the new object from reclaimed surplus heap memory.

3. (Previously presented) The method of claim 1, wherein reserving the guaranteed amount of heap memory from the common heap includes:

determining if there is sufficient heap memory available in the common heap; and

if not, performing garbage collection to reclaim allocated surplus heap memory, and

reserving heap memory for the task from reclaimed heap memory.

4. (Previously presented) The method of claim 1, wherein heap memory in the common heap is managed using a generational garbage collector.

5. (Original) The method of claim 4, wherein a plurality of tasks share an old generation of the generational garbage collector.

6. (Original) The method of claim 5, wherein each task of the plurality of tasks has a new generation of the generational garbage collector belonging to the task.

7. (Original) The method of claim 4, wherein the generational garbage collector is a copying garbage collector.

8. (Currently amended) A computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for managing heap memory in a multitasking system, the method comprising:

reserving a guaranteed amount of heap memory for a task from a common heap in the multitasking system, wherein the heap memory reserved for the task is separate from heap memory reserved for all other tasks in the common heap of the multitasking system;

receiving a request from the task to allocate heap memory for a new object; and

if heap memory is available in the guaranteed amount of heap memory for the task, allocating heap memory for the new object from the guaranteed amount of heap memory;

if heap memory is not available, checking if surplus heap memory is available in the common heap;

if surplus heap memory is available ~~so~~, reserving an additional amount of heap memory to the task from the common heap and allocating heap memory for the new object from the additional amount of heap memory, whereby allocating heap memory for the new object from the additional amount of heap memory delays garbage collection; and

if surplus heap memory is not available, performing garbage collection on the heap memory reserved for the task, wherein the task space is separate from all other task ~~space~~ spaces; ~~, whereby~~

wherein other tasks continue normal execution without interruption due to garbage collection~~, or subsequent memory compaction~~.

9. (Previously presented) The computer-readable storage medium of claim 8, wherein if surplus heap memory is not available in the common heap in addition to heap memory allocated to tasks, the method further comprises:

performing garbage collection on heap memory to reclaim unused reserved heap memory, and

allocating heap memory for the new object from reclaimed surplus heap memory.

10. (Previously presented) The computer-readable storage medium of claim 8, wherein reserving the guaranteed amount of heap memory from the common heap includes:

determining if there is sufficient heap memory available in the common heap; and

if not, performing garbage collection to reclaim allocated surplus heap memory, and

reserving heap memory for the task from reclaimed heap memory.

11. (Previously presented) The computer-readable storage medium of claim 8, wherein heap memory in the common heap is managed using a generational garbage collector.

12. (Original) The computer-readable storage medium of claim 11, wherein a plurality of tasks share an old generation of the generational garbage collector.

13. (Original) The computer-readable storage medium of claim 12, wherein each task of the plurality of tasks has a new generation of the generational garbage collector belonging to the task.

14. (Original) The computer-readable storage medium of claim 11, wherein the generational garbage collector is a copying garbage collector.

15. (Currently amended) An apparatus that facilitates managing computer heap memory in a multitasking system, comprising:

a computing device including a multitasking virtual machine;

a reserving mechanism within the multitasking virtual machine that is configured to reserve a guaranteed amount of physical heap memory for a task from a new generation space within a common heap in the multitasking system, wherein the heap memory reserved for the task is separate from heap memory reserved for all other tasks in the common heap of the multitasking system;

a receiving mechanism within the multitasking virtual machine that is configured to receive a request from the task to allocate heap memory for a new object;

the reserving mechanism that is further configured to reserve an additional amount of heap memory to the task from the common heap;

an allocating mechanism within the multitasking virtual machine that is configured to allocate heap memory for the new object from the guaranteed amount of heap memory; ~~memory~~

a reserving mechanism configured to reserve an additional amount of heap memory to the task from surplus memory within the common heap and allocating heap memory for the new object from the additional amount of heap memory, whereby allocating heap memory for the new object from the additional amount of heap memory delays garbage collection; and

a garbage collection mechanism configured to perform garbage collection on the heap memory reserved for the task, wherein the task space is separate from all other task ~~space~~ spaces; ~~whereby~~

wherein other tasks continue normal execution without interruption due to garbage collection~~, or subsequent memory compaction~~.

16. (Previously presented) The apparatus of claim 15, further comprising:

a garbage collecting mechanism that is configured to perform a garbage collection on heap memory to reclaim unused reserved heap memory, and

the allocating mechanism that is further configured to allocate heap memory for the new object from reclaimed heap memory

17. (Previously presented) The apparatus of claim 15, further comprising:

a determining mechanism that is configured to determine if there is sufficient heap memory available in the common heap; and

a garbage collection mechanism that is configured to perform a garbage collection to reclaim allocated surplus heap memory, and

the reserving mechanism that is further configured to reserve heap memory for the task from reclaimed heap memory.

18. (Original) The apparatus of claim 15, further comprising a generational garbage collector.

19. (Original) The apparatus of claim 18, wherein a plurality of tasks share an old generation of the generational garbage collector.

20. (Original) The apparatus of claim 19, wherein each task of the plurality of tasks has a new generation of the generational garbage collector belonging to the task.

21. (Original) The apparatus of claim 18, wherein the generational garbage collector is a copying garbage collector.